Citation

Conference on Programming Language Design and Implementation >archive

Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation >toc

1999 , Atlanta, Georgia, United States

A new framework for debugging globally optimized code

> Also published in ...

Authors

Le-Chun Wu
Rajiv Mirani
Harish Patil
Bruce Olsen
Wen-mei W. Hwu

Sponsors

SIGSOFT : ACM Special Interest Group on Software Engineering
SIGPLAN : ACM Special Interest Group on Programming Languages

Publisher

ACM Press New York, NY, USA

Pages: 181 - 191 Series-Proceeding-Article
Year of Publication: 1999
ISSN:0362-1340

doi> http://doi.acm.org/10.1145/301618.301663 (Use this link to Bookmark this page)

> full text > abstract > references > citings > index terms > peer to peer

> Discuss > Similar > Review this Article Save to Binder

> BibTex Format

↑ **FULL TEXT:** Access Rules

pdf 1.54 MB

↑ **ABSTRACT**

With an increasing number of executable binaries generated by optimizing compilers today,

h c g c cf c

providing a clear and correct source-level debugger for programmers to debug optimized code has become a necessity. In this paper, a new framework for debugging globally optimized code is proposed. This framework consists of a new code location mapping scheme, a data location tracking scheme, and an emulation-based forward recovery model. By taking over the control early and emulating instructions selectively, the debugger can preserve and gather the required program state for the recovery of expected variable values at source breakpoints. The framework has been prototyped in the IMPACT compiler and GDB-4.16. Preliminary experiments conducted on several SPEC95 integer programs have yielded encouraging results. The extra time needed for the debugger to calculate the limits of the emulated region and to emulate instructions is hardly noticeable, while the increase in executable file size due to the extra debug information is on average 76% of that of the executable file with no debug information.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 P. T. Zellweger, Interactive Source-Level Debugging of Optimized Programs. PhD thesis, Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, 1984.

2 P. T. Zellweger, "An interactive high-level debugger for control-flow optimized programs," SIGPLAN Notices, vol. 18, pp. 159-171, August 1983.

3 R. Gupta, "Debugging code reorganized by a trace scheduling compiler," Structured Programming, vol. 11, pp. 141- 150, July 1990.

4 Urs Hölzle , Craig Chambers , David Ungar, Debugging optimized code with dynamic deoptimization, Proceedings of the 5th ACM SIGPLAN conference on Programming language design and implementation, p.32-43, June 15-19, 1992, San Francisco, California, United States

5 John Hennessy, Symbolic Debugging of Optimized Code, ACM Transactions on Programming Languages and Systems (TOPLAS), v.4 n.3, p.323-344, July 1982

6 David Wall , Amitabh Srivastava , Fred Templin, A note on Hennessy's "symbolic debugging of optimized code", ACM Transactions on Programming Languages and Systems (TOPLAS), v.7 n.1, p.176-181, Jan. 1985

7 D. S. Coutant , S. Meloy , M. Ruscetta, DOC: a practical approach to source-level debugging of globally optimized code, Proceedings of the SIGPLAN'88 conference on Programming Language design and Implementation, p.125-134, June 20-24, 1988, Atlanta, Georgia, United States

8 Max Copperman, Debugging optimized code without being misled, California Institute of Technology, Pasadena, CA, 1993

9 Max Copperman, Debugging optimized code without being misled, ACM Transactions on Programming Languages and Systems (TOPLAS), v.16 n.3, p.387-427, May 1994

10 Roland Wismüller, Debugging of globally optimized programs using data flow analysis, Proceedings of the ACM SIGPLAN '94 conference on Programming language design and implementation, p.278-289, June 20-24, 1994, Orlando, Florida, United States

11 Ali-Reza Adl-Tabatabai , Thomas Gross, Evicted variables and the interaction of global register allocation and symbolic debugging, Proceedings of the 20th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.371-383, March 1993, Charleston, South Carolina, United

States

12 Ali-Reza Adl-Tabatabai , Thomas Gross, Detection and recovery of endangered variables caused by instruction scheduling, Proceedings of the conference on Programming language design and implementation, p.13-25, June 21-25, 1993, Albuquerque, New Mexico, United States

13 Ali-Reza Adl-Tabatabai , Thomas Gross, Source-level debugging of scalar optimized code, Proceedings of the ACM SIGPLAN '96 conference on Programming language design and implementation, p.33-43, May 21-24, 1996, Philadelphia, Pennsylvania, United States

14 A. Adl-Tabatabai, Source-Level Debugging of Globally Optimized Code. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1996.

15 L.-C. Wu and W. W. Hwu, "A new data-location tracking scheme for the recovery of expected variable values," Tech. Rep.IMPACT-98-07 (ftp: / / ftp. crhc. uiuc.edu / pub/IMPACT / report / im pact- 98- 07.dataloc.ps), IMPACT, University of Illinois, Urbana, IL, September 1998.

16 Alfred V. Aho , Ravi Sethi , Jeffrey D. Ullman, Compilers: principles, techniques, and tools, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1986

17 Pohua P. Chang , Scott A. Mahlke , William Y. Chen , Nancy J. Warter , Wen-mei W. Hwu, IMPACT: an architectural framework for multiple-instruction-issue processors, Proceedings of the 18th annual international symposium on Computer architecture, p.266-275, May 27-30, 1991, Toronto, Ontario, Canada

18 Caroline Tice , Susan L. Graham, OPTVIEW: a new approach for examining optimized code, Proceedings of the 1998 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering, p.19-26, June 16-16, 1998, Montreal, Quebec, Canada

↑ CITINGS

Clara Jaramillo , Rajiv Gupta , Mary Lou Soffa, Comparison checking: an approach to avoid debugging of optimized code, ACM SIGSOFT Software Engineering Notes, v.24 n.6, p.268-284, Nov. 1999

↑ INDEX TERMS

Primary Classification:

D. Software

↳ **D.3** PROGRAMMING LANGUAGES

↳ **D.3.3** Language Constructs and Features

↳ **Subjects:** Frameworks

Additional Classification:

D. Software

↳ **D.2** SOFTWARE ENGINEERING

↳ **D.2.5** Testing and Debugging

↳ **Subjects:** Debugging aids

↳ **D.3** PROGRAMMING LANGUAGES

↳ **D.3.4** Processors
↳ **Subjects:** Compilers

G. Mathematics of Computing
↳ **G.2** DISCRETE MATHEMATICS
↳ **G.2.2** Graph Theory
↳ **Subjects:** Path and circuit problems

General Terms:
Algorithms, Design, Experimentation, Languages, Measurement, Performance, Reliability, Theory

↑ **Peer to Peer - Readers of this Article have also read:**

- Editorial pointers
  Communications of the ACM 44, 9
  Diane Crawford
- News track
  Communications of the ACM 44, 9
  Robert Fox
- We Talk to Everybody
  Linux Journal 2000, 74es
  Marjorie Richardson , Jason Schumaker , David Penn
- Forum
  Communications of the ACM 44, 9
  Diane Crawford
- At the Forge
  Linux Journal 1998, 52es
  Reuven M. Lerner

↑ **This Article has also been published in:**

- **ACM SIGPLAN Notices**
  Volume 34 , Issue 5 (May 1999)

h c g c cf c